

In [6]:

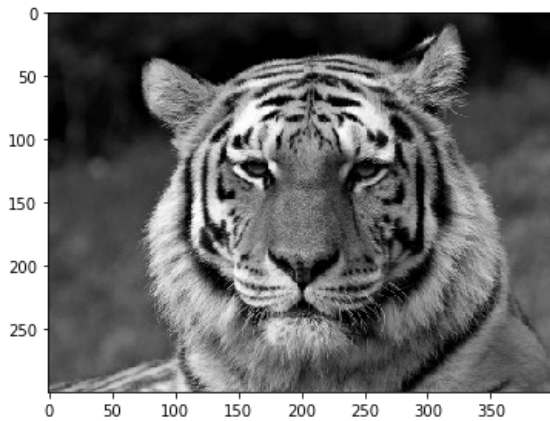
```
from skimage.io import imread, imshow
from skimage import img_as_float, img_as_ubyte, img_as_uint
from scipy.signal import convolve2d
%matplotlib inline
import numpy as np
```

In [7]:

```
img = imread('tiger-gray-small.png')
imshow(img)
```

Out[7]:

<matplotlib.image.AxesImage at 0x17da3d90>



In [8]:

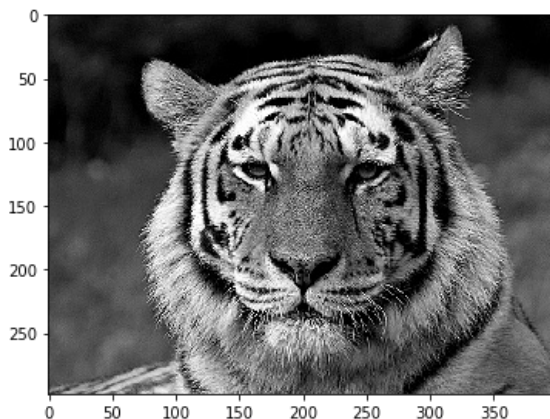
```
kernel = np.array([[ -1, -2, -1], [-2, 22, -2], [-1, -2, -1]]) # не забыть умножить на 1/10
kernel = kernel
def sharp_filter(img, kernel):
    img = img.copy()
    box_size = kernel.shape[0]
    filtered_img = np.zeros((img.shape[0] - box_size + 1, img.shape[1] - box_size + 1))
    for x in range(filtered_img.shape[0]):
        for y in range(filtered_img.shape[1]):
            filtered_img[x][y] = sum(sum(img[x:x + box_size, y:y + box_size] * kernel))/10
    return filtered_img
```

In [9]:

```
filtered_img = sharp_filter(img, kernel)
filtered_img = np.clip(filtered_img, 0, 255)
filtered_img = filtered_img.astype('uint8')
imshow(filtered_img)
```

Out[9]:

<matplotlib.image.AxesImage at 0x17dd7700>



In [10]:

```
print(filtered_img.shape)
```

(298, 398)

```
img1 = imread('unsharp-tiger.png')
print(img1.shape)
imshow("unsharp-tiger.png")
```

Out[11]:

```
print(img1[0,32], filtered_img[0,32])
for i in (img1 - filtered_img):
    print(i)
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

255

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]